Környezetfüggetlen és sztochasztikus nyelvtanok összehasonlítása többnyelvű gépi beszédfelismerési feladatban

Mozsolics Tamás, Tarján Balázs, Mihajlik Péter, Fegyó Tibor

Távközlési és Médiainformatikai Tanszék,
Budapesti Műszaki és Gazdaságtudományi Egyetem
{mozsolics, tarjanb, mihajlik, fegyo}@tmit.bme.hu

Kivonat: A szituációs beszédfelismerés egyik legfontosabb eleme a szituációhoz jól alkalmazkodó beszédfelismerő hálózat tervezése. Ezért megvizsgáltunk néhány hálózatépítési módszert, hogy összehasonlítsuk teljesítményüket. Az építés és tesztelés folyamatát összesen hat nyelven végeztük el: angol, francia, magyar, német, olasz és spanyol. Tesztelés céljából a telefonos hálózaton keresztül az utcáról vagy járműből rögzített, tájékozódási célú kérdésekből és kijelentésekből álló adatbázist használtunk. Magyar, német, olasz és spanyol nyelvekre összehasonlítottuk a fonéma- és grafémaalapú tervezési technikákat, s a magyar modellt különböző paraméterek változtatása mentén is vizsgáltuk. A hálózatokat saját fejlesztésű, WFST-s modellező rendszeren építettük, saját felismerőn futtattuk és HTK-val értékeltük ki.

1 Bevezetés

A TELEAUTO projekt célja egy olyan tájékozódási szolgáltatás biztosítása az autósok számára, ahol a gépkocsivezető szóban kérhet segítséget egy céllal kapcsolatban, ahová el szeretne jutni. A kéréseket egy külső helyszínen, egy kétszintes kiszolgáló rendszer várja, s rájuk első körben egy számítógép próbál válaszolni, s amennyiben ez sikertelen, akkor a gépi rendszer továbbküldi a kérést a diszpécsernek. Válaszként mindkét esetben a kívánt cél GPS- (Global Positioning System) koordinátáit kapja az autóban található navigációs eszköz vissza.

Ebben a cikkben mi a projekt gépi kiszolgáló moduljának tervezésével s megvalósításával foglalkozunk. A gépi modul lelke az automatikus beszédfelismerő szoftver (ASR: Automatic Speech Recogniser), mely úgy működik, hogy egy előre betöltött ún. beszédfelismerő hálózat mondataihoz illeszti a bejövő kérést, s ezek közül kiválasztja a legjobban illeszkedőt. A beszédfelismerő hálózat az adott szituációban általunk várt mondatok gyűjteménye, melyből a beszédfelismerő szoftver mindig egyet választ. Ahhoz, hogy a gépi alapú kiszolgálás hatékony legyen, alapvetően két dologhoz, az autó akusztikai környezetéhez és a beszédszituációhoz kell jól alkalmazkodni. Ezek közül az első jelfeldolgozási, konkrétan szűrési, a második pedig a beszédfelismerő hálózat építéséhez kapcsolódó feladat. A cikk ez utóbbit tárgyalja.

2 Nyelvi modell építése

A kitűzött feladatból látható, hogy a projektben található gépi beszédfelismerőtől nem várjuk el, hogy adott nyelven bármit megértsen, sőt azt sem, hogy pontosan megértse a kérés minden részletét, elég, ha a célpontot jól megérti. Nincs szükség általános szöveget leíró beszédfelismerő hálózatra, sőt egy adott szituációra optimalizált modell sokkal hatékonyabb lehet. Mivel a téma elég speciális, általában nem áll rendelkezésre adatbázis a szituációban előforduló mondatokról, így nekünk kell azt összegyűjteni minden nyelvre.

2.1 Szituációhoz tartozó mondatok gyűjtése

Nézzünk meg néhány célpontkeresésre irányuló, várható példamondatot magyar nyelvre:

„Hol van a közelben McDonald's?”
„Hol van egy könyvesbolt?”
„Hol lehet egy OTP-automata?”
„Hol van a közelben kórház?”
„Hol található a közelben Tesco?”
„Hol van a Holokauszt Múzeum?”

Elméletileg persze végtelenféle mondat lehet, s biztosan nem tudjuk összegyűjteni mindet, de minél többet sikerül, annál jobb lesz a modell. Szerencsére az a megfigyelés, hogy az előforduló kérések szerkezete független a konkrét célponttól jelentősen csökkenti a variáció számát, hisz nem kell az összes *Hol van?* típusú kérdést felsorolni, s így időt, munkát, memóriát spórolhatunk. Ketté kell tehát választani a gyűjtési feladatot tipikus mondatszerkezetek és célpontok gyűjtésére. Ennek tükrében az előző példa mondatszerkezetei, illetve célállomásai (a [cél] változó) így festenek:

„Hol van a közelben [cél]?”	*„McDonald's/POI”*
„Hol van egy [cél]?”	*„könyvesbolt/POI”*
„Hol lehet egy [cél]?”	*„OTP/POI automata/POI”*
„Hol van a közelben [cél]?”	*„kórház/POI”*
„Hol található a közelben [cél]?”	*„Tesco/POI”*
„Hol van a [cél]?”	*„Holokauszt Múzeum/POI”*

Ez a fajta szétválasztás több szempontból előnyös. Egyrészt megkönnyíti a tervezést és bővítést, másrészt a külön célállomás (POI: Point Of Interest) lista kompatibilis a diszpécserközpont adatbázisával, mivel gépi felismeréskor lényegében csak a célpontok pontos felismerésére törekszünk, ezen célok szavai könnyen felcímkézhetők (l. /POI címkék a célpontok szavai után), megkönnyítve a gépi felismerést, lényegkiemelést. Ekkor egy *Hol van a közelben McDonald's/POI?* típusú találatból könnyedén kiemelhető a célállomásra vonatkozó rész, illetve ezen részek jelenlétében/hiányában könnyedén eldönthető, hogy lehet-e az adott gépi felismerésnek használható eredménye-e vagy sem. A gyakorlatban a nyelvi modellek ehhez a szituációhoz lényegében az itt leírt módon készültek, annyi különbséggel, hogy a mondatszerkezetek egy hatékonyabb leíró formátumban, a Phoenix cég Parser nevű rendsze-

rében (l. [6]) definiált GRA formátumban lettek a mondatszerkezetek definiálva, illetve a célpontlista, több különálló kategóriára lett bontva, pl.: bevásárlás, szállók, étkezdék. Készítsünk az előző példánkból GRA-modellt. Ehhez először rendezzük egymás mellé a hasonló mondat szerkezeteket, s ebből a GRA-modell:

„Hol van a [cél]?”
„Hol van a közelben [cél]?”
„Hol van egy [cél]?”
„Hol lehet egy [cél]?”
„Hol található a közelben [cél]?”

```
[varhato_keres]
    (Hol VAN NEVELO [cel]?)
VAN
    (van)(lehet)(található)
NEVELO
    (a)(a közelben)(egy) ;
```

A GRA-modellben a [ ] zárójelbe tett kifejezés a makró definíciót jelent, a makró *„ ; ”* jellel zárul. Az egyes makrókban definiálhatunk változókat a változatok leírására, ezeket csupa nagybetűvel írjuk, s még a makrón belül kifejtjük. A makrók, illetve változók által leírt egyes változatok ( ) zárójelbe kerülnek. A *„ *”* jel, azt jelenti, hogy *„vele vagy nélküle”*, tehát mindkét eset előfordulhat. A GRA formátum láthatóan nem támogatja az ékezetes karakterek használatát változó- és makródefiníciók esetén.

2.2 A CFG és az N-gram modellek

A CFG (Context Free Grammar) lényegében a 2.1. bekezdésben leírt módon összegyűjtött mondatszerkezetekből épített aciklikus modell, melybe behelyettesítjük a POI-listát.

Az N-gram modell a CFG-től eltérően már ciklikus felépítésű, melynek átmeneti valószínűségei ebben az esetben a CFG-ben összegyűjtött mondatokból mint tanítószövegből lettek tanítva egy simítási eljárást követően. Mindezt a CMU Logios nevű szoftverével végeztük el.

Ez a modell szerkezetéből adódóan elvileg toleránsabb az adott szituáció olyan nem várt mondataival szemben, ahol a várt szavak szerepelnek ugyan, de a várttól kissé eltérő sorrendben és/vagy kissé eltérő mondathossz mellett.

2.3 Emberi nyelv – gépi nyelv

Mi, emberek a beszédről szóegységekben gondolkodunk, s így a mondandónkat szósorozatok formájában fogalmazzuk meg. Ez számunkra természetes, így modelljeinket is szó-, vagy morfémaalapon készítjük. Viszont a számítógép szóalapú hálózathoz nem tud pontosan illeszteni. Ennek oka, hogy a szó mint alapegység számát és hosszát tekintve túl variábilis, illetve egy-egy mondat relatíve kevés szóból, illetve morfémából épül fel.

A beszédfelismerésre használható hatékony gépi modell – a beszéd sztochasztikus jellegéből adódóan – valószínűség-számítás alapú, s rejtett Markov-modellek (HMM: Hidden Markov Model) az elemei. Ilyen szövegkörnyezetben az illeszkedés jelentése, hasonló valószínűségi paramétervektorok birtoklása. Mivel az emberi nyelv, avagy a hálózattervezés hatékony szintje (szavak szintje), s a gépi nyelv, avagy a gépi beszédfelismerés hatékony szintje (HMM-ek szintje) nem esik egybe, ezért a megtervezett hálózatainkat át kell vinni a HMM-szintre a felismerő szoftverbe töltést megelőzően.

Ez a transzformáció három lépésben végezhető el, melyeket fonetikus átírásnak, környezetfüggősítésnek, illetve nyelvimodell-beillesztésnek nevezik. Trigráf alatt – a trifón analógiára – a szomszédjaitól mint környezettől függő grafémát értjük.

$$\text{szó sorozat} \rightarrow \left\{ \begin{array}{lcl} \text{fonéma sorozat} & \rightarrow & \text{trifón sorozat} \\ \text{graféma sorozat} & \rightarrow & \text{trigráf sorozat} \end{array} \right\} \rightarrow \text{HMM sorozat}$$

1. ábra. A nyelvi modellezés 4 szintje.

3 WFST Framework

A WFST Framework egy olyan egységes matematikai keretrendszer, melynek elemei speciális, címkézett és súlyozott irányított gráfok ún. WFST-k (Weighted Finite State Transducer) s a rajtuk végezhető műveletek. Ebben a matematikai modellben a hálózatoptimalizálás és a 2.3. bekezdésben látott szintlépések is elvégezhetők, l. [1, 2].

Egy beszédfelismerő hálózat mint WFST szerkezetileg optimális, ha determinisztikus, vagyis egy adott bemeneti sorozat egyértelműen meghatározza, hogy merre menjünk benne, minimális, vagyis a lehető legtömörebben épített és súlyaiban sztochasztikus, vagyis egynemű, kiugró értékektől mentes. A beszédfelismerő hálózat építését teljes egészében a 2. ábrán látható műveletsor írja le.

$$\underbrace{\underbrace{\underbrace{\underbrace{\text{H o wpush(min(det(C o det(L o } \underbrace{\text{G}}_{\text{szó szintű modell}} \text{))))}_{\text{fonetikus szintű modell}}}_{\text{trifón szintű modell}}}_{\text{HMM szintű modell}}$$

2. ábra. A nyelvi modelljeinkhez használt nyelv független WFST-műveletsor.

Ahol, a G (Grammar) transzducer a szószintű nyelvi modellünk, az L (Library) a nyelvi modellben szereplő szavak fonetikus vagy grafémás átiratait tartalmazó szótár, a C az ún. környezetfüggősítő (Context-Dependency) transzducer és a H (HMM-Library) az adott nyelv trifónjainak/trigráfjainak HMM-es átiratait tartalmazó szótár. Az *o* jelöli az ún. kompozíció műveletet, mellyel az egyes szintlépések elvégezhetők. A *det* gráfok determinizálásához hasonlóan a WFST egy olyan átépítését jelenti olyan ekvivalens WFST-vé, melyben a bemeneti szimbólumsorozatnak megfelelő haladás mindig egyértelmű. Sajnos ez a determinizált tulajdonságú ekvivalens WFST-k esetében nem mindig létezik, l. [5]. A *min* alatt itt olyan műveletek csoportját értem, melylyel az eredeti WFST-vel ekvivalens tömörebb WFST állítható elő. Ez lehet gráfminimalizáció és címkéket okosabban rendező/összevonó algoritmus is. A *wpush* a súlyok egyenletesebb eloszlását biztosítja a WFST-nkben.

A 2. ábrán látható műveletsor fontos tulajdonsága, hogy nyelvfüggetlen, ezért csak a H, C, L és G transzducereket kell előállítanunk minden nyelvre. Aciklikus nyelvi

modell esetén (pl.: CFG) a fonémaszintű modell determinizációja minimalizációval helyettesíthető a még tömörebb hálózat érdekében. Mivel az [1, 2] irodalmak egyértelműen leírják, hogy kell a H, C és L transzducereket felépíteni, ezért koncentrálunk mi is elsősorban ebben a cikkben a G nyelvi modell építésének ismertetésére.

4 Nyelvfüggő kihívások

Annak ellenére, hogy a 3. fejezetben megmutattuk, hogy a modellezési módszer nyelvfüggetlen, a tudásforrások összeállításánál akadnak nyelvfüggő részproblémák, mint pl.: helyhatározó és tárgyragok a magyarban, vagy a különböző nemű szavak névelői a németben.

4.1 Hely-és tárgyragok a magyar nyelvben

A TELEAUTO-s szituációra összegyűjtött mondatszerkezetek java részében a célpontok, tárgyként vagy helyhatározóként szerepelnek, hisz valamelyik *áruház**hoz***, *reptér**re***, *mozi**ba*** mennénk, s *éjjel-nappali**t*** keresünk.

Az természetesen ésszerűtlen elvárás lenne, hogy több tízezer POI-nak összes ragozott alakját kézzel állítsuk elő. Szerencsére ezen ragok (tipikusan a *-t*, *-ba/be*, *-ra/re* és *-hoz/hez/höz*) megfelelő alakjának kiválasztása jól automatizálható. Tekintsük át pl. a *-ba/be* ragok közötti választásra megadott alábbi szabályokat:

```
;a magánhangzók osztályozása
mghL == a á o ó u ú; mély
mghH == e é i í ö ő ü ű
            ; magas
;a ba/be ragozás szabályai
;1. hasonulások
e[ba] = é b e;
mghL-e[ba] = é b a;
mghL,msh-e[ba] = é b a;
a[ba] = á b a;
az[ba] = a b a;
ez[ba] = e b e;
```

```
;2. az utolsó magánhangzó dönt
  mghL-[ba] = b a;
  mghL,msh-[ba] = b a;
  mghL,msh,msh-[ba] = b a;
  mghH-[ba] = b e;
  mghH,msh-[ba] = b e;
  mghH,msh,msh-[ba] = b e;
  ;3.a korábbi mélymagánhangzó dominanciája
  mghL,mghH-[ba] = b a;
  mghL,mghH,msh-[ba] = b a;
  mghL,msh,mghH-[ba] = b a;
 mghL,msh,mghH,msh-[ba] = b a;
```

melyek közül a szabályok hosszúságuk szerinti sorrendben kerülnek sorra, tehát a szoftverek először mindig a hosszabbakat próbálják illeszteni. A módszer elve, hogy a magánhangzókat két osztályra, mély és magas magánhangzókra (mghL és mghH változók), s a szavak *-ba/be* ragot közvetlen megelőző része alapján (– előtti rész) hoz döntést. Nézzünk meg néhány példát:

```
mozi[ba] = mghL,msh,mghH-[ba] = b a
pizzéria[ba] = a[ba] = á b a
IKEA[ba] = a[ba] = á b a
parkoló[ba] = mghL-[ba] = b a
Debrecen[ba] = mghH,msh-[ba] = b e
edzőterem[ba] = mghH,msh-[ba] = b e
Allee[ba] = mghL,msh-e[ba] = é b a
```

Ezt a megoldást a magyar nyelvi modellekhez készített szótárakból egy 266 szavas részszótáron teszteltük, s csak 15-ször hibázott, ami kb. 95% pontosságnak felel meg.

5 Tesztelés

A beszédfelismerő hálózat építéséhez és teszteléséhez használt források fontosabb adatai kiolvashatók az 1. táblázatból.

A hat nyelven készített modellek építésekor nyelvenként átlagosan körülbelül 25000 mondat szerkezetet sikerült összegyűjteni/generálni. Az összehasonlítós tesztek nagyjából 500 POI-val készültek, átlagosan 900 szavas teljes szótárméret mellett. A teszteléshez összesen 56 beszélő által felmondott, nagyjából 500 felvétel állt rendelkezésünkre. Ezek mindegyike valós körülmények között, gépkocsiban felvett, telefonos beszélgetés minőségű felvétel volt. A felvételeket nyelvenként számos, különböző nemű, korú, natív beszélőtől rögzítettünk.

5.1 Felismerési pontossági jellemzők

A tesztelés táblázataiban tipikusan az alább ismertetett 5 paraméter jelenik meg.

Az S_{ACC}, illetve $S_{ACC,POI}$ azt mutatja meg, hogy a mondatok, illetve az POI-kkal kapcsolatos mondatrészek hány százalékát sikerült hibátlanul felismerni. Hasonlóan értelmezhetők a W_{ACC}, illetve $W_{ACC,POI}$ paraméterek az elhangzott szavakra.

Azt, hogy a felismerési idő hogyan arányul a tesztszöveg időbeli hosszához, mutatja meg az RTF, az ún. Real-Time Factor, tehát pl. ha RTF=0.2, akkor az elhangzási idő ötöde kell a feldolgozáshoz.

1. táblázat: A teszteléshez használt modellek és felvételek főbb paraméterei.

nyelvi modell	angol	francia	magyar	német	olasz	spanyol	összes
mondatszerkezetek	36746	4066	18782	68296	5425	17604	25153
POI-k száma	501	501	518	519	506	529	512
szótárméret	624	727	1886	677	689	748	892
átírás adatai	angol	francia	magyar	német	olasz	spanyol	átlag
fonémák száma	44	35	38	39	57	25	39,66
grafémák száma	--	--	33	31	29	33	31,50
akuszt. modell	angol	francia	magyar	német	olasz	spanyol	átlag
tanítószöveg[óra]	17,8	57,9	28,9	62,1	93,7	56,5	52,81
tesztfelvételek	angol	francia	magyar	német	olasz	spanyol	összes
száma	57	40	266	26	70	28	487

5.2 Hardver- és szoftverkörnyezet

2. táblázat: A fonetikus átírásokhoz használt szoftverek és szótárak 6 nyelvre.

angol	francia	magyar	német	olasz	spanyol
M-Phon	Liaphon	M-Phon	Txt2pho	M-Phon	Txt2pho

A teszteléshez használt hardver egy T7300 nevű Core2Duo architektúrájú, 2 GHz-es processzorú laptop volt, 2 GB RAM-mal, 32 bites Windows operációs rendszer alatt. A beszédfelismerő hálózatokat a saját készítésű M-System nevű szoftverrendszerrel építettük, s a szintén saját VOXerver nevű felismerőn futtattuk le, majd az eredményeket HTK-val (l. [4]) értékeltük ki. A fonetikus átíráshoz a 2. táblázatban látható szoftvereket és szótárakat használtuk.

5.3 Teszttípusok

5.3.1 CFG vs. N-gram

Ebben a tesztben fonémaalapú CFG- és N-gram-modelleket hasonlítottunk össze 6 különböző nyelven. Mivel a modellek nagyjából hasonló kondíciók mellett 500 POI készültek, ez a teszt alkalmas a nyelvek osztályozására is e feladat kapcsán. A teszt eredményeit a 3. táblázatban foglaltuk össze.

Ha a két modellt akarjuk egymáshoz hasonlítani, akkor elsősorban a táblázat utolsó oszlopát érdemes megvizsgálni, ugyanis itt szerepelnek az eredményeknek a felvételek számával súlyozott átlagai. Ezek alapján elmondható, hogy a két megoldás között nincs különösebben nagy különbség. Az N-gram-megoldás a fő paraméterekben $W_{ACC,POI}$ és $S_{ACC,POI}$ 1-1.5%-al túlteljesíti a CFG-t pontosságban, cserébe a feldolgozási idő kb. 28%-kal nő.

Ha a különböző nyelvek egymáshoz képesti viszonyát tekintjük, akkor az angol az egyetlen, melynek paraméterei kb. 10%-nál jobban eltérnek a többitől. Ennek okai egyrészt a rendelkezésre álló relatíve kisebb adatbázis (l. 1. táblázat), másrészt az angol nyelv beszélt és írott alakja közötti hatalmas eltérés, mely a fonetikus átírás során nehezen leküzdhető feladat elé állítja a mérnököket.

3. táblázat: A fonémaalapú CFG- és N-gram-modellek eredményei 6 nyelvre.

CFG	angol	francia	magyar	német	olasz	spanyol	átlag
W_{ACC} [%]	55,16	73,86	78,16	68,71	73,11	79,78	73,98
$W_{ACC,POI}$ [%]	46,53	79,34	74,68	84,78	76,81	80,60	72,95
S_{ACC} [%]	16,07	40,00	57,42	34,62	37,14	46,43	46,39
$S_{ACC,POI}$ [%]	36,36	72,50	72,33	76,92	71,43	82,14	68,81
RTF [$*t_{valós}$]	0,344	0,108	0,164	0,191	0,072	0,140	0,167
N-gram	angol	francia	magyar	német	olasz	spanyol	átlag
W_{ACC} [%]	51,92	71,59	76,74	71,78	73,37	84,70	73,12
$W_{ACC,POI}$ [%]	48,51	75,21	75,78	84,78	80,43	86,57	74,31
S_{ACC} [%]	14,29	37,50	52,36	26,92	31,43	53,57	42,39
$S_{ACC,POI}$ [%]	36,36	72,50	72,98	73,08	77,14	78,57	69,58
RTF [$*t_{valós}$]	0,508	0,170	0,171	0,290	0,117	0,266	0,214

5.3.2 Szituációs modellek vs. POI-lista

Egy jogosan felmerülő kérdés lehet, hogy mennyivel kapnánk rosszabb eredményt, ha egyszerűen a POI-listából, a szituációhoz alkalmazkodó mondatszerkezetek nélkül építenénk fonémaalapú beszédfelismerő hálózatot.

Összevetettük hát ezt a szólistás hálózatot a korábbi két modellel magyar nyelv esetében, s ez a teszt a 4. táblázatban látható eredményeket hozta. Ha kiolvassuk a fő paraméterek ($W_{ACC,POI}$ és $S_{ACC,POI}$) értékeit, láthatjuk, hogy a szituációkhoz alkalmazkodó mondatszerkezetek alkalmazásával kb. 2.5-3-szor pontosabb felismerés lehetséges.

4. táblázat: A POI-listából épített fonémaalapú modell összevetése a CFG- és N-gram-modellekkel, magyar nyelvre.

CFG	szólista	CFG	N-gram
W_{ACC} [%]	12,18	78,16	76,74
$W_{ACC,POI}$ [%]	24,27	74,68	75,78
S_{ACC} [%]	00,00	57,42	52,36
$S_{ACC,POI}$ [%]	30,08	72,33	72,98
RTF [$*t_{valós}$]	0,201	0,164	0,171

5.3.3 Fonémás vs. grafémás

Ezek a tesztek csak négy nyelven (magyar, olasz, német és spanyol) készültek, mivel a grafémás modellek csak olyan nyelvek esetében működnek jól, ahol a kiejtés és az írott alak között elég szoros kapcsolat van.

Az 1. táblázatból látható, hogy a nyelveknek átlagosan kisebb a grafémakészlet nagysága, mint a fonémáké, ezért a grafémás modellektől tömörebb felépítést s valamivel pontatlanabb nyelvleírást, s ezáltal némileg pontatlanabb felismerési eredményeket vártunk.

A POI-k vonatkozásában sok külföldi eredetű szó lehetséges egy adott nyelvterületen, pl. a magyar modell teszteléséhez használt tesztfelvételeink esetében a 412 POI-hoz kapcsolódó szavunkból 55 (kb. 15%) volt külföldi (pl.: Erste, McDonald's, Renault), ezért a fonéma- és grafémaalapú modellek összehasonlítása csak úgy lehetett fair, ha mindkét esetben kivételszótárat használunk a külföldi eredetű szavakra.

Ezen szótárak közötti eltérés csak annyi, hogy míg a fonémaalapú megoldásnál az idegen szavak magyar kiejtése, addig a grafémaalapú megoldásnál azoknak megfelelő grafémás alak szerepel a kivételek között pl.: a *„Rossmann”* szó átiratai

```
Rossmann = r o sz m a n ;    //fonema alapu kivetel szotar
Rossmann = r o s s z m a n ; //grafema alapu kivetel szotar
```

magyar modellek esetében.

5. táblázat: A grafémaalapú CFG-modellek eredményei 4 nyelvre.

CFG	angol	francia	magyar	német	olasz	spanyol	átlag
W_{ACC} [%]	--	--	77,17	63,80	77,28	79,78	76,49
$W_{ACC,POI}$ [%]	--	--	72,02	82,61	84,78	83,58	75,85
S_{ACC} [%]	--	--	59,40	23,08	38,57	39,29	51,80
$S_{ACC,POI}$ [%]	--	--	72,08	73,08	77,14	78,57	73,52
RTF [$*t_{valós}$]	--	--	0,137	0,235	0,092	0,163	0,137

A teszt során kapott eredmények az 5. táblázatban láthatók. A nyelvek közül kettő (magyar és német) eredményei valamivel rosszabbak, a spanyolé nagyjából egyforma, az olaszé pedig meglepő módon jóval jobb lett, mint a fonémás változaté. Ez utóbbi oka valószínűleg a saját kezűleg összegyűjtött olasz átírási szabályok hiányosságaiban keresendő.

A két módszer átlagos pontosságáról a 6. táblázat alapján elmondható, hogy az nagyjából egyforma, s a fonetikus megoldás előnye feltehetően egy jobb olasz fonetikus átírási megoldás mellett sem több néhány %-nál.

Kivételszótárat mindenképpen ajánlott használni grafémás esetben is, hiányukban az adott 4 nyelvre nagyjából 10%-os idegen szó arány mellett a tesztfelvételeken átlagosan 5%-kal rosszabb $S_{ACC,POI}$, $W_{ACC,POI}$ eredményt kaptunk a fonémás megoldáshoz képest.

6. táblázat: A graféma- és fonémaalapú modellek súlyozott átlaga 4 nyelvre.

Felvételek számával súlyozott átlagok	grafémás CFG	fonémás CFG
W_{ACC} [%]	76,49	76,74
$W_{ACC,POI}$ [%]	75,85	76,16
S_{ACC} [%]	51,80	51,47
$S_{ACC,POI}$ [%]	73,52	73.18
RTF [$*t_{valós}$]	0,137	0,148

5.3.4 A felismerési pontosság függése a POI-k számától

Fontos kérdés, hogy a POI-k számának növelésével, hogyan alakulnak a felismerési pontossági értékek. Nyilvánvalóan csökkeni fognak, hisz a POI-k növekedésével egyre több hasonló nevű célpontot kapunk, melyek között egyre nehezebb választani. Nem mindegy azonban, hogy ez a csökkenés milyen függvény szerint és milyen ütemben történik. Ennek vizsgálatához az 5.3.1. bekezdés fonetikus magyar CFG-modelljének POI-készletét bővítettük 5 lépésben egészen 5000-ig.

7. táblázat: A felismerési pontosság jellemzőinek alakulása az $500<N_{POI}<5000$ tartományban, magyar CFG-modell esetén.

POI-k száma	500	1000	1500	3000	4000	5000
W_{ACC} [%]	77,28	76,71	76,63	75,18	73,41	72,35
$W_{ACC,POI}$ [%]	72,41	71,03	69,77	67,09	65,47	61,95
S_{ACC} [%]	56,25	54,90	54,90	53,54	49,21	46,85
$S_{ACC,POI}$ [%]	70,75	69,29	68,90	67,33	65,60	63,05

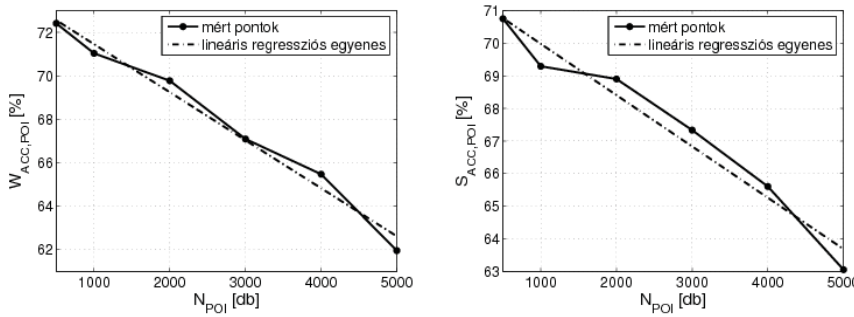
A tesztek eredményei kiolvashatók a 7. táblázatból. Az itt látható 4 pontossági paraméter közül minket leginkább a $W_{ACC,POI}$ és $S_{ACC,POI}$ alakulása érdekel, hisz az előbbi azt adja meg, hogy az esetek hány százalékában találunk a POI-hoz kapcsolódó szavakat, illetve hány százalékban találjuk meg az egyes POI-khoz tartozó összes szót. Ezen jellemzők alakulásának könnyítése végett, az $500<N_{POI}<5000$ tartományban a mért értékeket analitikus tesztfüggvényekkel közelítettük regressziót alkalmazva. A lineáris regressziós tesztfüggvények (lásd a 4. ábrát) mindkét esetben, pontosságban felülteljesítették az exponenciális, illetve hatványkitevős társaikat, ezért kijelenthető, hogy a vizsgálati tartományban az N_{POI} értékének növelésével a $W_{ACC,POI}$ és

$S_{ACC,POI}$ felismerési pontosságok egyenletes ütemben csökkennek. A közelítő egyenesek pontos egyenletei megtalálhatók a 8. táblázatban. Ezen egyenletek alapján a POI-k számának 1000-rel növelése 2,21% $W_{ACC,POI}$, illetve 1,57% $S_{ACC,POI}$ csökkenéssel jár.

Mondhatjuk persze, hogy még 5000 POI sem túl sok, s hogy az autók navigációs rendszere ennél nyilván többet tartalmaz. Ennek ellenére a POI-k száma jól kordában tartható, ha minden esetben az autótól egy bizonyos hatósugarú körben levő célokra szűkítjük a keresést.

8. táblázat: A $W_{ACC,POI}$ és $S_{ACC,POI}$ felismerési pontosságot közelítő lineáris regressziós egyenesek egyenletei az $500<N_{POI}<5000$ tartományban, magyar CFG-modell esetén.

	regressziós egyenes egyenlete
$W_{ACC,POI}$ [%]	$73.6610 - 0.002209 \cdot N_{POI}$
$S_{ACC,POI}$ [%]	$71.5529 - 0.001574 \cdot N_{POI}$



3. ábra. A $\mathbf{W_{ACC,POI}}$ és $\mathbf{S_{ACC,POI}}$ felismerési pontosságot közelítő lineáris regressziós egyenesek az $500<N_{POI}<5000$ tartományban, magyar CFG modell esetén.

5.3.5 A felismerési hiba szórása az egyes szituációkban

Az 5.3.1-5.3.4. bekezdésekben kaptunk átlagos felismerési pontossági adatokat egy-egy adott nyelvre készített adott típusú modell esetén. Eddig arról viszont nem volt szó, hogy mekkora eltérések lehetnek egy-egy eltérő szituációban (eltérő jel-zaj viszony, zavaró környezeti zajok, pl. mentőautó hangja) ezen átlagértékektől.

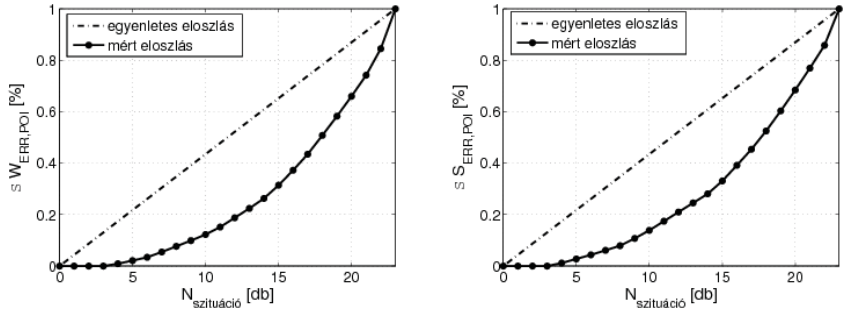
Ennek szemléltetéséhez a fonémaalapú magyar CFG-modellt szituációnként is leteszteltük. Az eredetileg 27 felvételsorból összevontuk az 5 legkisebbet, melyekhez 5-nél kevesebb felvétel tartozott, s előállítottuk 23 különböző szituáció átlagos $W_{ERR,POI}$ ($=1-W_{ACC,POI}$) és $S_{ERR,POI}$ ($=1-S_{ACC,POI}$) paramétereit.

9. táblázat: A fonémaalapú magyar CFG-modell 23 különböző szituáció alapján számított statisztikus paraméterei.

	átlag	terjedelem	szórás
$W_{ERR,POI}$ [%]	26,51	100	20,56
$S_{ERR,POI}$ [%]	28,63	100	21,44

Ezen tesztelés a 9. táblázatban összefoglalt eredményei alapján elmondható, hogy a $W_{ERR,POI}$ és $S_{ERR,POI}$ paraméterek a [0%,100%] tartományban mozogtak, átlagos értékük (ahogy azt már korábbról is ismertük) 26.5-28.5% volt, s az egyes szituációkban ettől 20.5-21.5%-kal tértek el átlagosan.

A koncentráltságról a 4. ábrán látható eloszlási függvények segítségével tájékozódhatunk. Az ábrán az átlót követő függvény képviseli az egyenletes hibamegoszlást az egyes szituációk között, az alattuk futó konvex alakú függvények pedig a mért eloszlás függvények. Ezek alapján pl. a 23-ból a 6 legrosszabb eredmény felelős az összes $W_{ERR,POI}$ és $S_{ERR,POI}$ hibák 50%-áért, a 7 legrosszabb pedig a hibák 60%-áért.



4. ábra. Az összes hiba eloszlása a fonémaalapú magyar CFG-modell 23 különböző szituációjában.

5.3.6 A felismerési pontosság függése a mondatszerkezetek számától

Miután az 5.3.2. bekezdésben láthattuk, hogy hasznos, hogy POI-listánkat az adott szituációhoz illeszkedő mondatszerkezetekbe illesztjük, érdemes lenne megvizsgálni, hogy ezen szerkezetek variációinak száma, hogy hat a felismerési pontosságra.

A 10. táblázatból látható, hogy a mondatszerkezetek számának, s ezáltal közvetve a találati arány csökkenésével az N-gram felismerési pontosság előnye, rugalmas szerkezeti felépítésének köszönhetően, a korábbi 1-2%-ról 5-6%-ra emelkedett.

10. táblázat: A felismerési pontosság alakulása a mondatszerkezetek számának csökkenésével fonémaalapú magyar CFG és N-gram esetén.

fonéma CFG				
rel. mondatszerkezetszám	1,000	0,841	0,682	0,540
$W_{ACC,POI}$ [%]	72,41	71,90	55,22	45,38
$S_{ACC,POI}$ [%]	70,75	70,36	56,18	42,74
fonéma N-gram				
rel. mondatszerkezetszám	1,000	0,841	0,682	0,540
$W_{ACC,POI}$ [%]	73,18	70,08	61,44	49,41
$S_{ACC,POI}$ [%]	70,97	68,13	60,24	47,22

6 Összefoglalás

A TELEAUTO projekt beszédalapú tájékozódást segítő szolgáltatás készítését tűzte ki céljául autóvezetők számára, növelve ezzel biztonságukat és komfortérzetüket. Segítségkéréskor egy külső központot hívhatunk, s egyszerű hétköznapi kérésekért cserébe az autónk navigációs rendszere a kívánt cél koordinátáit kapja vissza.

Ennek a rendszernek fontos eleme egy automatikus beszédfelismerő rendszer, mely jó esetben jelentősen csökkenti a diszpécserek munkamennyiségét, azáltal, hogy a beérkező kérdések jelentős hányadát megválaszolja. Mivel a felismerő szoftver a beérkezett kéréseket egy előre betöltött beszédfelismerő hálózathoz hasonlítja, a megoldás kulcsa ennek a hálózatnak a tervezésében rejlik.

Bemutattuk a WFST Framework nevű matematikai modellt, s annak keretein belül történő beszédfelismerő hálózatépítés előnyeit, mely szerint egyszerű, nyelvfüggetlen megoldást kapunk az emberi és gépnyelv közötti alapegység transzformációra szavakról egészen a HMM-ekig, illetve a beszédfelismerő hálózat optimalizálására is.

Szemléltettük a magyar nyelvben a tárgy- és helyragok illesztésének problémáját, mint egy nyelvfüggő kihívást, s hatékony automatikus módszert adtunk a megoldásához.

Ismertettünk két nyelvi modell típust, a CFG-t és az N-gram-ot. Megmutattuk, hogy mindkét megoldás hasonlóan jó, az N-gram kicsit több erőforrásért cserébe kicsit pontosabb eredményt ad, főleg abban az esetben, ha az eredeti várakozásoktól eltérő hosszúságú vagy szerkezetű kérdésekre kell válaszolni. Tervezett hálózatoknak mindkét megoldásnál két fő eleme van. Egyrészt az elérni kívánt pontok (POI-k) listája, melynek bővítése a felismerési pontosságot lineáris ütemben csökkenti, illetve a szituációhoz illeszkedő mondatszerkezetek, melyek hiányában a hibák száma drasztikusan nőne.

Bemutattuk és teszteltük a grafémaalapú tervezést mint a fonéma alapú rendszer alternatíváját olyan nyelvekre, ahol az írott és beszélt nyelv kapcsolata szoros. Ez kis hibanövekedés mellett rengeteg energiát spórol meg, melyet nyelvenként a külső fonetikus átíró szoftverek tesztelésére és rendszerbe illesztésére, vagy átírási szabályok gyűjtésére és tesztelésére fordítanánk. Fontos, hogy a grafémaalapú módszerrel modellezett nyelvekhez illeszkedő kivételszótárakat alkalmazzunk a modellezett nyelvhez képest idegen POI-vonatkozású szavakra.

A különböző nyelvekre készített modellek teszteléskor láthattuk, hogy az adott 6 nyelvre, az angolt kivéve, megoldásunk egyformán jól működik. Remélhetőleg a jövőben a rendelkezésünkre álló, az akusztikus modelltanításra használható adatbázis bővülésével az angol modellünk eredményei is felzárkóznak a többiéhez.

Köszönetnyilvánítás

Ez a kutatás az OM-00102/2007 számú "TELEAUTO" projekt keretén belül készült.

Bibliográfia

1. Szarvas, M.: Efficient Large Vocabulary Continuous Speech Recognition Using Weighted Finite-state Transducers – The Development of a Hungarian Dictation System. PhD Thesis, Department of Computer Science, Tokyo Institute of Technology, Tokyo (2003)
2. Mohri, M., Pereira, F. C. N., Riley, M.: Speech recognition with weighted finite-state transducers. In: Rabiner, L., Juang, F. (szerk.): Handbook on Speech Processing and Speech Communication, Part E: Speech recognition. Springer-Verlag, Heidelberg, Germany (2008)
3. Magyar Telefonos Beszéd Adatbázis: http://alpha.tmit.bme.hu/speech/hdbMTBA.php
4. Young, S., Ollason, D., Valtchev, V., Woodland, P.: The HTK book. (for HTK version 3.4) (2009) http://htk.eng.cam.ac.uk
5. Allauzen, C., Mohri, M.: Efficient algorithms for testing the twins property. Journal of Automata, Languages and Combinatorics Vol. 8 No.2 (2003) 117–144
6. Center for Spoken Language Research of Colorado: Phoenix parser for spontaneous speech. http://cslr.colorado.edu/~whw/phoenix/